BMC Bioinformatics

Research article

DNA microarray data and contextual analysis of correlation graphs

Jacques Rougemont* and Pascal Hingamp

Address: TAGC, INSERM-ERM 206, Parc Scientifique de Luminy Case 906, 13288 Marseille Cedex 09, France

Email: Jacques Rougemont* - rougemont@tagc.univ-mrs.fr; Pascal Hingamp - hingamp@tagc.univ-mrs.fr

* Corresponding author

Published: 29 April 2003

BMC Bioinformatics 2003, **4**:15

Received: 23 December 2002

Accepted: 29 April 2003

This article is available from: http://www.biomedcentral.com/1471-2105/4/15

Abstract

Background: DNA microarrays are used to produce large sets of expression measurements from which specific biological information is sought. Their analysis requires efficient and reliable algorithms for dimensional reduction, classification and annotation.

Results: We study networks of co-expressed genes obtained from DNA microarray experiments. The mathematical concept of curvature on graphs is used to group genes or samples into clusters to which relevant gene or sample annotations are automatically assigned. Application to publicly available yeast and human lymphoma data demonstrates the reliability of the method in spite of its simplicity, especially with respect to the small number of parameters involved.

Conclusions: We provide a method for automatically determining relevant gene clusters among the many genes monitored with microarrays. The automatic annotations and the graphical interface improve the readability of the data. A *C++* implementation, called *Trixy*, is available from http://tagc.univ-mrs.fr/bioinformatics/trixy.html.

Background

Measurements of gene expression levels by microarray experiments create a high-throughput of data, the interpretation of which increasingly requires novel and efficient dimensionality reduction strategies. Many clustering methods have been proposed (see for example [1–5] and the more comprehensive reviews [6,7]) and are widely used. These algorithms group genes and/or samples into clusters of similar expression profiles, in order to suggest possible functional relationships between them. The importance of graphical representations and of automatic cluster annotations stands out from many recent publications [1,8–12] devoted to gene functions prediction, prognosis or diagnosis of cancer subtypes for instance.

Similar problems arise in the analysis of large interaction networks [13–18] where one tries to extract sub-networks satisfying some significance criteria. The problem of finding web pages dedicated to the same topic is an example that will appeal to the experience of every reader (in this case the network's nodes are the URLs, with HTML links).

We propose a new method which combines one of these network analysis techniques with the classical correlation-based clustering for studying DNA microarray data. It provides a novel graphical representation, a cluster forming rationale and cluster annotations through correlation with gene or sample keywords. The algorithm relies on only two user-controlled parameters, therefore sensitivity of the results to a particular choice of parameters can be checked effectively.

The algorithm is based on the notion of curvature introduced in [13] (this is the same as the clustering coefficient of [19]), which we apply to the network of co-expressed genes where nodes are genes (or samples) and links

symbolize co-expression. We define clusters as connected regions of the graph with high curvature, which is the local density of triangular relations. The gene or sample clusters are the densest regions of the corresponding correlation graph, which we will show has biological relevance as intuitively expected. We must emphasize that curvature is typically extremely low in random graphs that have small average degree compared to the number of nodes (which is usually the case in biological networks [14,20]). Clusters of high curvature are thus highly non-random structures.

We have implemented these concepts in the freely available program *Trixy*. It is a graphical interface for visualising the graph, the clusters and the automatic annotations providing a straightforward tool for exploring microarray data. The *C++* source code and sample *Perl* parsers are freely available from http://tagc.univ-mrs.fr/bioinformatics/trixy.html. We also provide the data files adapted from the original yeast [1] and lymphoma [21] sets as examples. We have compiled and used the program on both Linux and Windows platforms. Compiling on other platforms has not been attempted but is theoretically possible.

On the performance side, clustering and display with *Trixy* requires CPU time and memory size comparable to hierarchical clustering as performed in [1].

Algorithm

Curvature on Graphs

The discussion below focuses on the problem of clustering genes. The symmetric question of clustering samples can be treated similarly.

A DNA microarray data set consists of expression levels of N genes in M different experimental conditions (M different RNA samples). This is organised in an $N \times M$ matrix $X_{i,j}$, $i = 1,...,N$; $j = 1,...,M$ each row of which contains the expression profile of a given gene across all samples. We are interested in patterns of co-expression, namely groups of genes with parallel or anti-parallel profiles. We measure co-expression of genes g_k and $g\ell$ by the (Pearson) correlation $\mathrm{cor}(k, \ell)$ between their profiles:

$$\mathrm{cor}(k,\ell) = \frac{\sum_{j=1}^{M}\left(X_{k,j} - \mu_k\right)\cdot\left(X_{\ell,j} - \mu_\ell\right)}{\sigma_k \sigma_\ell}, \qquad (1)$$

where μ_i and σ_i denote the mean and the standard deviation of row i. This creates a correlation matrix which is an $N \times N$ symmetric matrix (because $\mathrm{cor}(k, \ell) = \mathrm{cor}(\ell, k)$).

We construct a correlation graph as follows. We first make a node n for each gene. We then choose a threshold $T_{\mathrm{cor}} \in [0,1]$ and draw a link between genes g_k and $g\ell$ if $\mathrm{cor}(k, \ell) \geq T_{\mathrm{cor}}$. This can be understood as follows: a graph with N nodes is defined by its adjacency matrix A (the $N \times N$ matrix such that $A_{i,j} = 1$ if i and j are joined, 0 otherwise [22]). We obtain A from the correlation matrix by binarisation: we replace $\mathrm{cor}(k, \ell)$ by 0 if it lies between $-T_{\mathrm{cor}}$ and T_{cor} and by 1 otherwise.

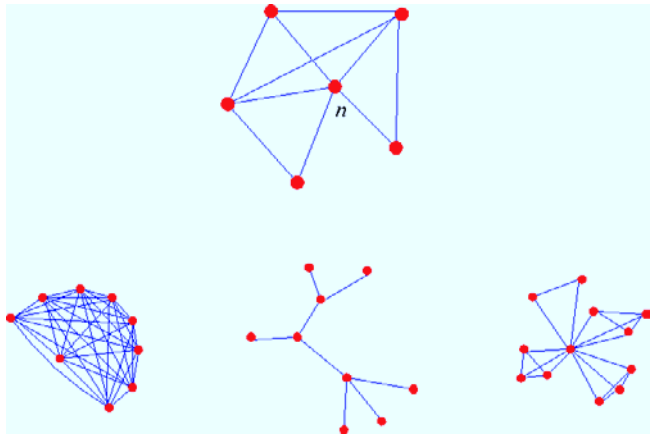

Figure 1
Top: the node n has $v = 5$ neighbours and $t = 5$ triangles, thus curvature curv(n) = 1/2, see Eq. (2). Bottom left: a complete graph, each node has curvature 1. Center: a tree, each node has curvature 0 or undefined. Right: the central node is a hub with curvature $\approx 1/v$

We next introduce the concept of curvature (or clustering coefficient) on a graph [13,19]. Each node n has a curvature which is a function of the number v of neighbours (nodes to which it is linked) and the number t of triangles (pairs of adjacent neighbours, see Figure 1), given by the formula

$$\mathrm{curv}(n) = \frac{t}{v(v-1)/2}. \qquad (2)$$

Remark that $v(v - 1)/2$ is the maximum number of triangles that can be drawn on v neighbours hence curv(n) lies between 0 and 1 if $v > 1$ and is undefined otherwise (see Figure 1 for examples of graphs and curvature).

There is a natural notion of distance between nodes in a graph [22]: it is the number of links in the shortest path connecting them (distance is infinite if there is no such path). Let $d_n(i, j)$ be the distance between the ith and jth neighbours of n: either $d_n(i, j) = 1$ (these two neighbours are linked) or $d_n(i, j) = 2$ (they are not, the shortest path goes through n). A simple computation shows that

$$\mathrm{curv}(n) = \frac{\sum_{0<i<j\le v} \left(2 - d_n(i,j)\right)}{\sum_{0<i<j\le v} 1} = 2 - \langle d_n \rangle, \qquad (3)$$

where $\langle d_n \rangle$ is the average distance between pairs of neighbours of n. Hence one can picture high curvature as high local density (low average distances).

Given a curvature threshold $T_{\mathrm{curv}} \in [0; 1]$ we select a subgraph by deleting all nodes with a curvature below T_{curv} as well as their links. This splits the graph into several connected components. Each such connected component will be interpreted as a coherent cluster of co-expressed genes (see Figure 2). Varying T_{curv} adds or removes nodes and links and thus modifies the clusters displayed (possibly merging or splitting some of the clusters). The reader can think of the graph as a sea bed, curvature being height. The curvature threshold T_{curv} is the sea level and the clusters are the emerging islands. Changing the correlation threshold T_{cor} changes the landscape while changing the curvature threshold T_{curv} only moves the sea level up and down.

Consider the Internet analogy of a University web site: the index page has many links to all department's web pages. It is unlikely that, for instance, the biology department's page provides a link to the literature department's page. Therefore, the index page will surely have a small curvature (few of its neighbours have links between them). However, the home page of the biology department has external links to biology departments in other universities with which it has common interests. These other pages will certainly also have external links to many of the same pages, again because they share similar interests. Therefore, a cluster of high curvature will emerge, comprising all the biology departments web pages. This reasoning applies to virtually any communication network and we demonstrate below that it can also be usefully applied to correlation graphs of gene expression profiles.

The program *Trixy* implements the algorithm described above in a user-friendly graphical interface. It is written in *C++* using the free *Qt* graphical library. It uses embedded *Perl* for parsing data inputs, which has the advantage that loading data saved under a new format only requires rewriting a *Perl* script which can be picked at run time. We have mostly used *Trixy* for clustering genes, but sample clustering can also be performed simply by using a modified parser which rotates the matrix. Similarly, an appropriate *Perl* script could simply fetch gene annotations from web servers such as http://www.geneontology.org/ rather than read them from a local file.

Normalisation and Parameters

A few simple data processing tools are provided in *Trixy*: log transform, samples centering (by subtracting the mean or the median) and samples reduction (division by the standard deviation). After these operations have been performed, the correlation matrix is computed and the curvature of each node is deduced from it. At this point, the user can view (using the "Eisengram" standard colour representation of the matrix, such as Figure 4) or save the resulting data set as a flat file. The graph is then built and displayed (as in Figure 2). Although the correlation threshold T_{cor} is set before loading the data, the curvature threshold T_{curv} can be varied as the graph is displayed. Starting from an initially high value of T_{curv} and lowering it progressively unveils new nodes and new clusters. It increases the size of existing clusters, sometimes merging several of them (Figure 2). This gives a feeling for the robustness of the clustering and for the closeness of clusters.

Our advice for the choice of T_{cor} is to set it to a value which retains only links significantly stronger than expected by pure chance (this depends on the particular data set and can be determined by bootstrapping, see *e.g.* [5]). The parameter T_{curv} is different. We have observed that the best value is often cluster-dependent. We have a more dynamic view on this parameter: the way clusters change as T_{curv} moves is informative. A good way of picking the best threshold is by maximising the annotation scores (see below).

Automatic Annotation

Trixy allows the user to provide annotation files for sample and genes. They consist of a list of keywords associated with each of the gene and/or sample names.

On the one hand, a cluster of genes can be associated with an over-represented gene keyword by giving a score to each annotation equal to its frequency in the cluster.

On the other hand, for sample annotations, a correlation score is computed. Suppose a cluster consists of genes $g_1, \ldots, g_K$. For each sample keyword W, we create a discriminating vector g_0 which takes the value 1 on each sample associated with W and -1 otherwise. The annotation score is the average absolute correlation with keyword W: $\frac{1}{K}\sum_{k=1}^{K}\left|\mathrm{cor}(0,k)\right|$.

Both scores yield numbers between 0 and 1, the closer to 1 the more significant the annotation. We discard annotations that were not present for at least 10% of the samples and 2 of the genes in each cluster.

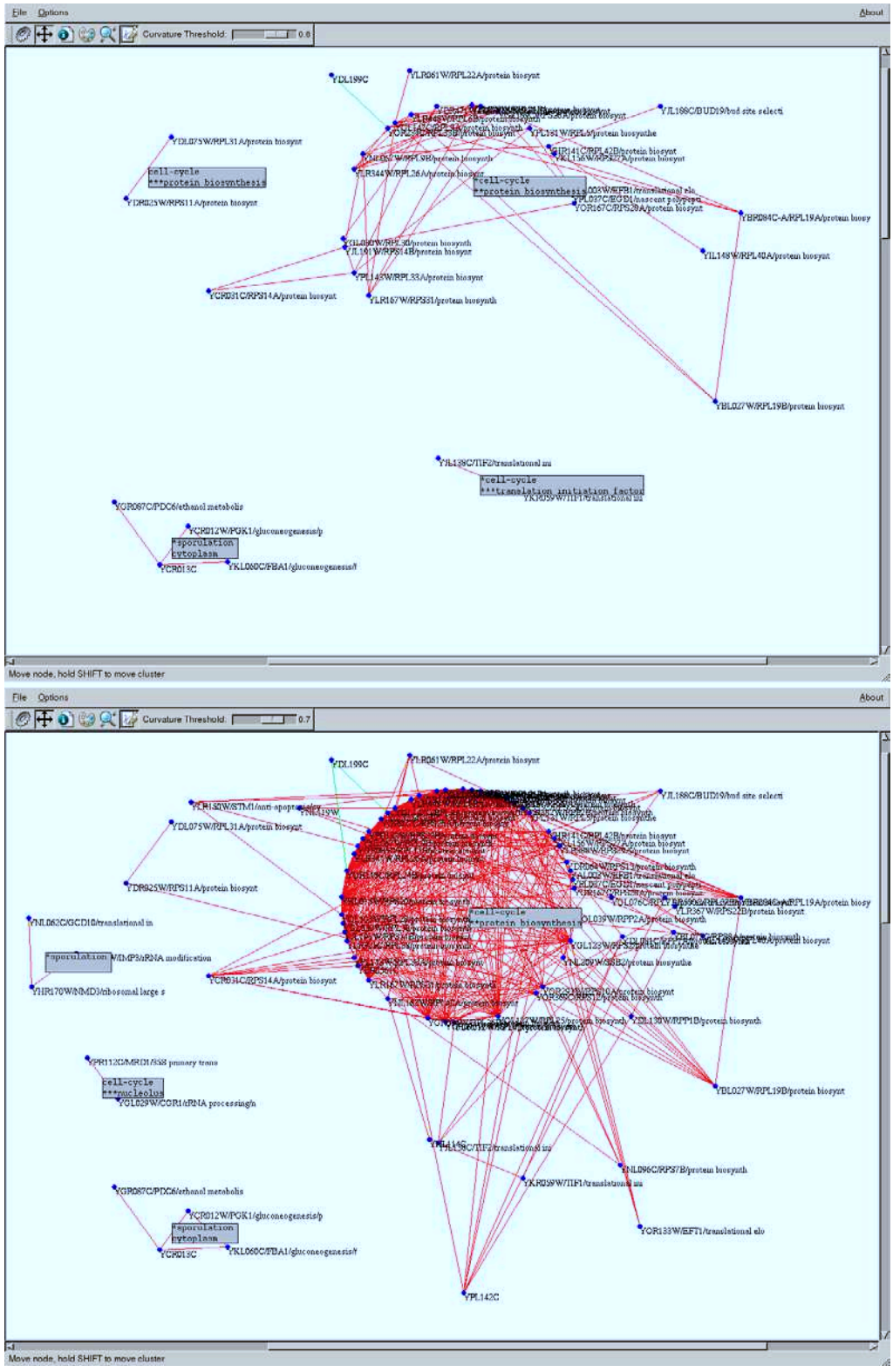

Figure 2
Screenshots of the main window of *Trixy* displaying a part of the yeast gene expression graph with T_{cor} = 0.85. Top: T_{curv} = 0.80, bottom: T_{curv} = 0.70. Grey boxes display automatic annotations with sample keyword on the first line and gene keyword on the second

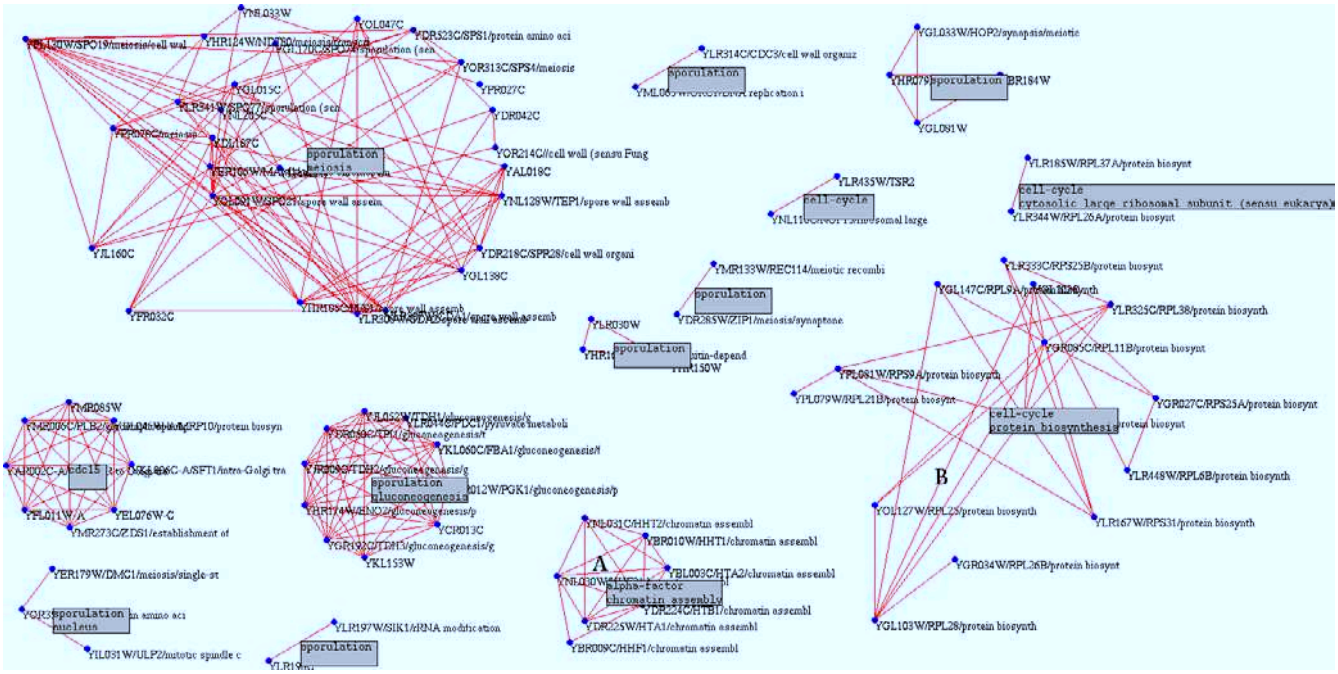

Figure 3
The full graph based on *Saccharomyces cerevisiae* 6221 gene expression profiles across 80 experiments with thresholds T_{curv} = 0.70 at T_{cor} = 0.90

Visualisation

The graph is displayed with a different colour code for links representing positive or negative correlations (in Figure 2, negative correlations are shown in green, see also Figure 7).

Each cluster can be selected and the corresponding data subset viewed (as a colour-coded table such as Figure 4). If annotations were provided, those with the highest scores are listed and the cluster can be saved as a data file, gene list or colour picture.

Results

Yeast Gene Expression Data

We have applied our algorithm to the data set of gene expression of the budding yeast Saccharomyces cerevisiae available from the website http://rana.lbl.gov/EisenData.htm and described in [1]. We have used Gene Ontology gene annotations from the Saccharomyces Genome Database (SGD) [23]. The sample keywords were extracted from the original expression data file and in this case do not yield interpretable annotations (see the lymphoma section below for a more convincing example of the usefulness of sample annotations).

Even with threshold values as high as T_{cor} = 0.90 and T_{curv} = 0.70, we get clearly delineated clusters (see Figure 3). Note that only 263 out of a total of 6221 genes have positive curvature at this value of the correlation threshold, yielding 2075 links.

Most of the clusters obtained appear biologically coherent. For example the *chromatin assembly* cluster contains all the 9 histone genes for T_{cor} = 0.80 and T_{curv} = 0.64 (Figure 3, cluster A: it only shows 7 of the genes at this level of correlation). It is disconnected from the rest of the graph and extremely robust with respect to changes in the parameters. The *ubiquitin dependent protein catabolism* (Figure 4) cluster appears at a much lower curvature but is extremely coherent with 17 out of 17 proteolysis genes. For the sake of comparison, a proteasome cluster of similar size obtained using hierarchical clustering contains 3 genes unrelated to proteolysis.

Much larger clusters are also visible, such as *the protein biosynthesis* cluster (Figure 3, cluster B and Figure 2). However, it is very sensitive to variations of the thresholds and can include over 900 genes.

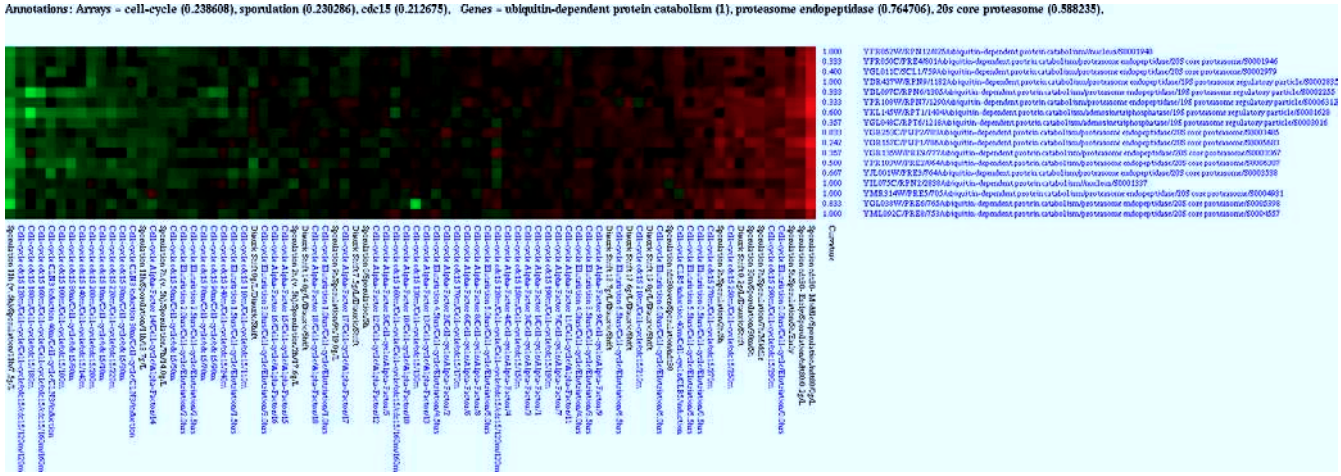

Figure 4
The *ubiquitin dependent protein catabolism* cluster of yeast gene expression at T_{cor} = 0.80 and T_{curv} = 0.18

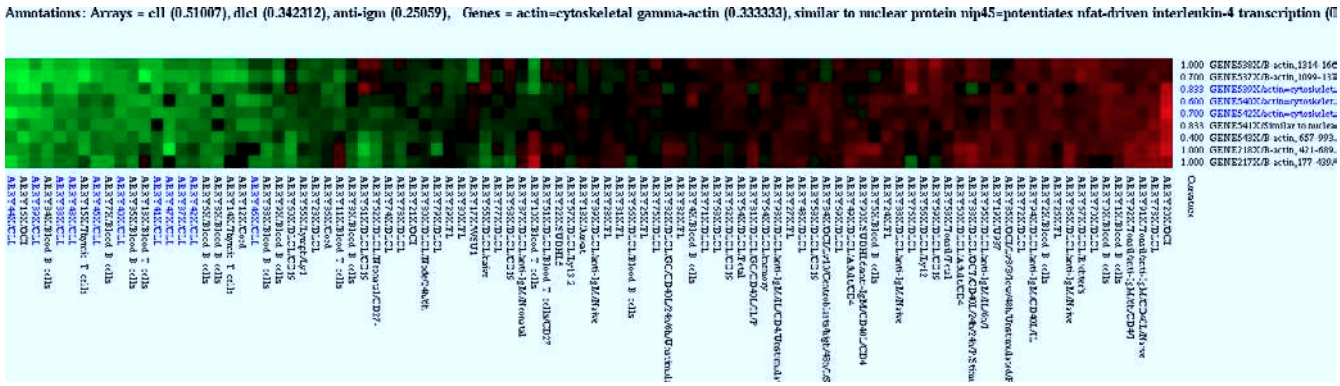

Figure 5
A cluster of genes under-expressed in the *CLL* tumors obtained from the lymphoma data with T_{cor} = 0.80 and T_{curv} = 0.40

B-cell Lymphoma

We have also used the data set of the lymphoma study available from the website http://llmpp.nih.gov/lymphoma/ and published in [21]. Sample and gene annotations were extracted from the names included in the data file. We were in particular interested in the classification of the tumor subtypes called chronic lymphocytic leukaemia (CLL) and diffuse large B-cell lymphoma (DLCL).

Using a threshold T_{cor} = 0.80 we obtain nicely discriminating clusters, for example Figure 5 which separates the *CLL* tumors and Figure 6 which sorts the *DLCL* tumors. The latter cluster eventually merges with a cluster of *Interferon-induced* genes (data not shown) as the curvature threshold decreases: they are like two hills on the same island.

We also have a good example of a property that is often observed with our graphical representation: negative correlations are much rarer than positive ones and are carried by just a few nodes, which are almost certainly represser genes. Figure 7 shows a small part of the graph where 7 nodes are mostly anti-correlated with the rest of a cluster of 307 genes. However, let us emphasize that the graphs shown here do not represent gene interaction networks *per se*, they are merely a means of clustering genes co-expressed within the selected samples.

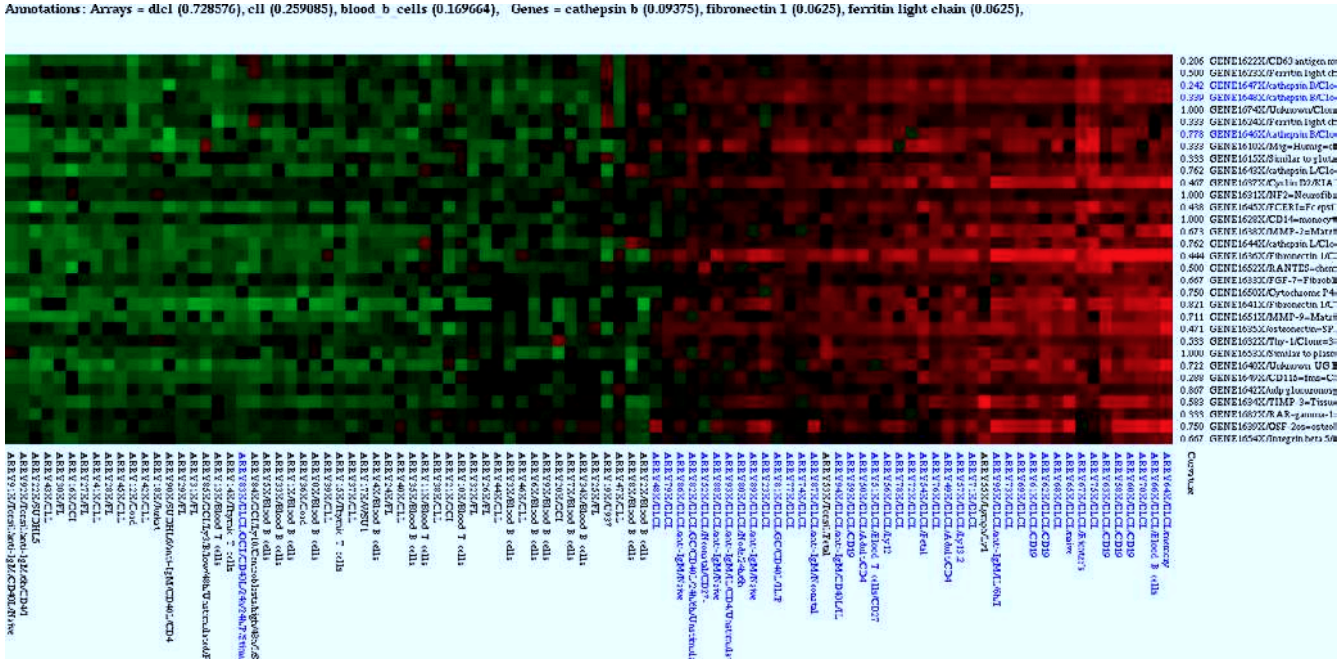

Figure 6
A cluster of genes over-expressed in the *DLCL* tumors obtained from the lymphoma data with T_{cor} = 0.80 and T_{curv} = 0.24

This data set provides a good test bed for sample clustering. In this case, the graph's nodes are the samples and links denote a correlation between samples (columns of the expression level matrix *X*). As shown in Figure 8, the result is clear cut: all clusters are associated with a single sample subtype, be it *B-cells*, *T-cells*, *FL* (follicular lymphoma) or *CLL*.

Statistical Validation

As stated earlier, mathematical theory shows that random graphs have a very low number of triangles [14,20]. We can check this statement numerically by using random permutations of a real data set. We have randomly reordered the yeast data, gene by gene (a random permutation in each line), computed the curvature of each node, and repeated this operation 1000 times. At T_{cor} = 0.6 we deduced a probability of positive curvature of 5×10^{-5}, the probability of having a degree larger than 1 was 2×10^{-4} (maximum degree observed: 6). At T_{cor} = 0.7 the probability of positive curvature was 10^{-6}. The comparison with the curvature distribution of the real biological data is displayed in Figure 9.

Statistical validation of an annotation by a particular keyword can be performed with similar methods. For example in the case of the clusters shown in Figure 5,6, we have performed 10,000 random permutations of the sample keywords. We have obtained a maximum annotation score of 0.315 for the *CLL* cluster and the mean score plus two standard deviations was equal to 0.177, whilst the annotation score of the original cluster was 0.510. Similarly for the *DLCL* cluster, the maximum score after 10, 000 permutations was 0.367, the mean score plus two standard deviations 0.172, whilst the original score was 0.729.

Permutations of gene keywords can be computed explicitely. For example 66 of the 6221 yeast genes have the GO annotation *ubiquitin dependent protein catabolism*. The probability of having 17 of them in the same cluster of size 17 (see Figure 4) is of the order of 10^{-35}.

Discussion and Conclusions

We have described an algorithm for visualising and analysing large microarray data sets. It combines traditional correlation distances and new graph-theoretical ideas. We have implemented this algorithm in a convenient graphical interface and evaluated its performance on well established data sets.

Curvature thresholds split the graph into clusters which appear to be biologically meaningful. An automatic annotation procedure associates keywords with clusters, which are consistent with previous publications [1,21].

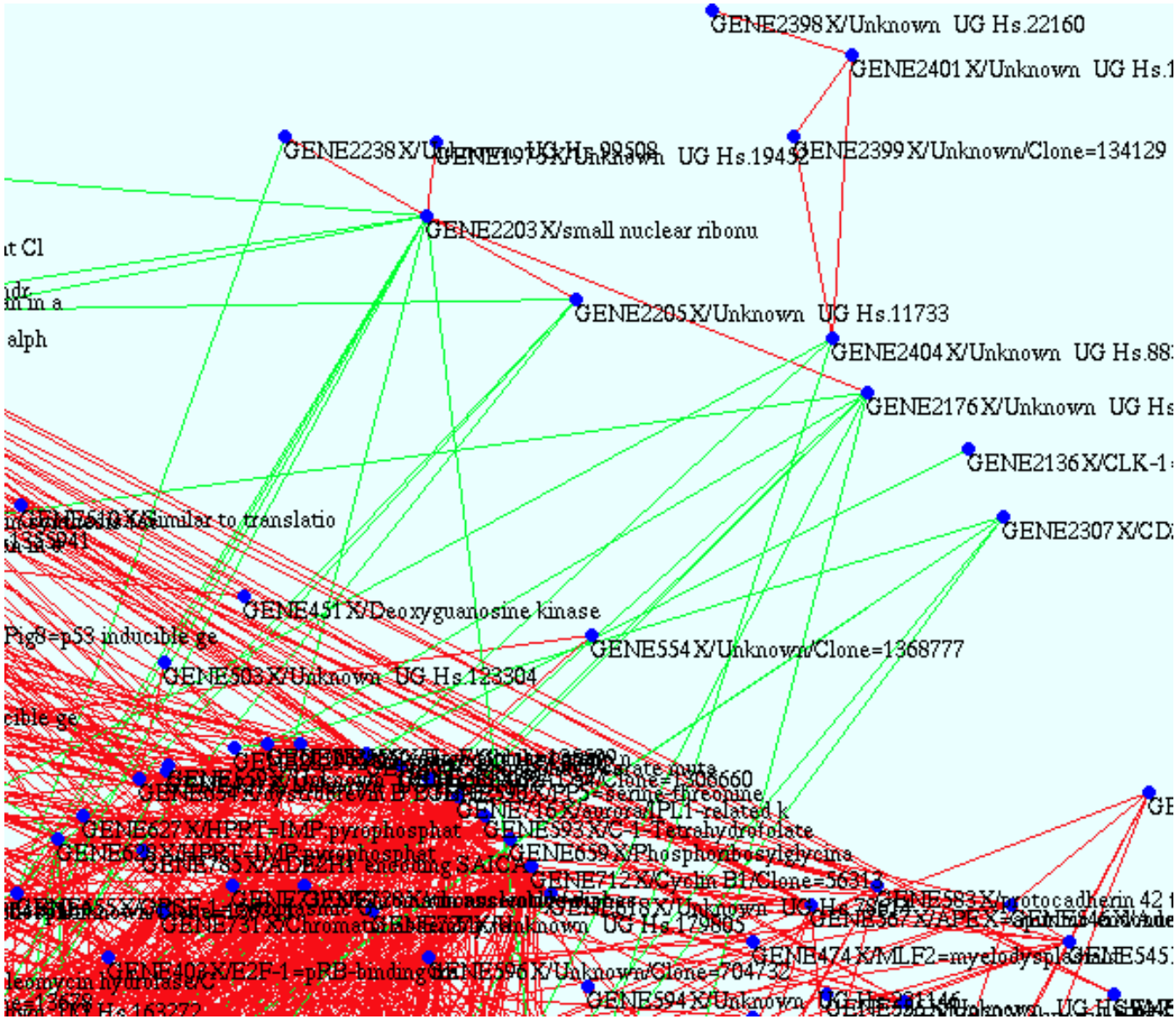

Figure 7
A small part of the lymphoma graph showing putative repressor genes (green links are negative correlations). Threshold values are T_{cor} = 0.70 and T_{curv} = 0.45

Our approach uses a local analysis of the correlation graph as opposed to global properties such as small-world [19] or scale-free properties [24]. Hubs [25] are not seen as relevant in our approach since they usually make little contribution to the information carried by the graph (like index pages in the worldwide web). The closest approach to ours is that of [14,15] in the sense that triangles are network motifs, but again, we only draw conclusions from their local density not their frequency at the level of the whole graph. This also emphasizes the difference between our method and the more commonly used hierarchical clustering [1]: because we focus on triangular relations (rare motif in graphs) rather than simple links (very common motif), we obtain a drastic dimensional reduction (see Figure 3) whereas hierarchical clustering retains all the data and does not in itself delineate clusters. Furthermore the stronger constraint offered by triangular links as opposed to single link methods ensure more coherent clusters.

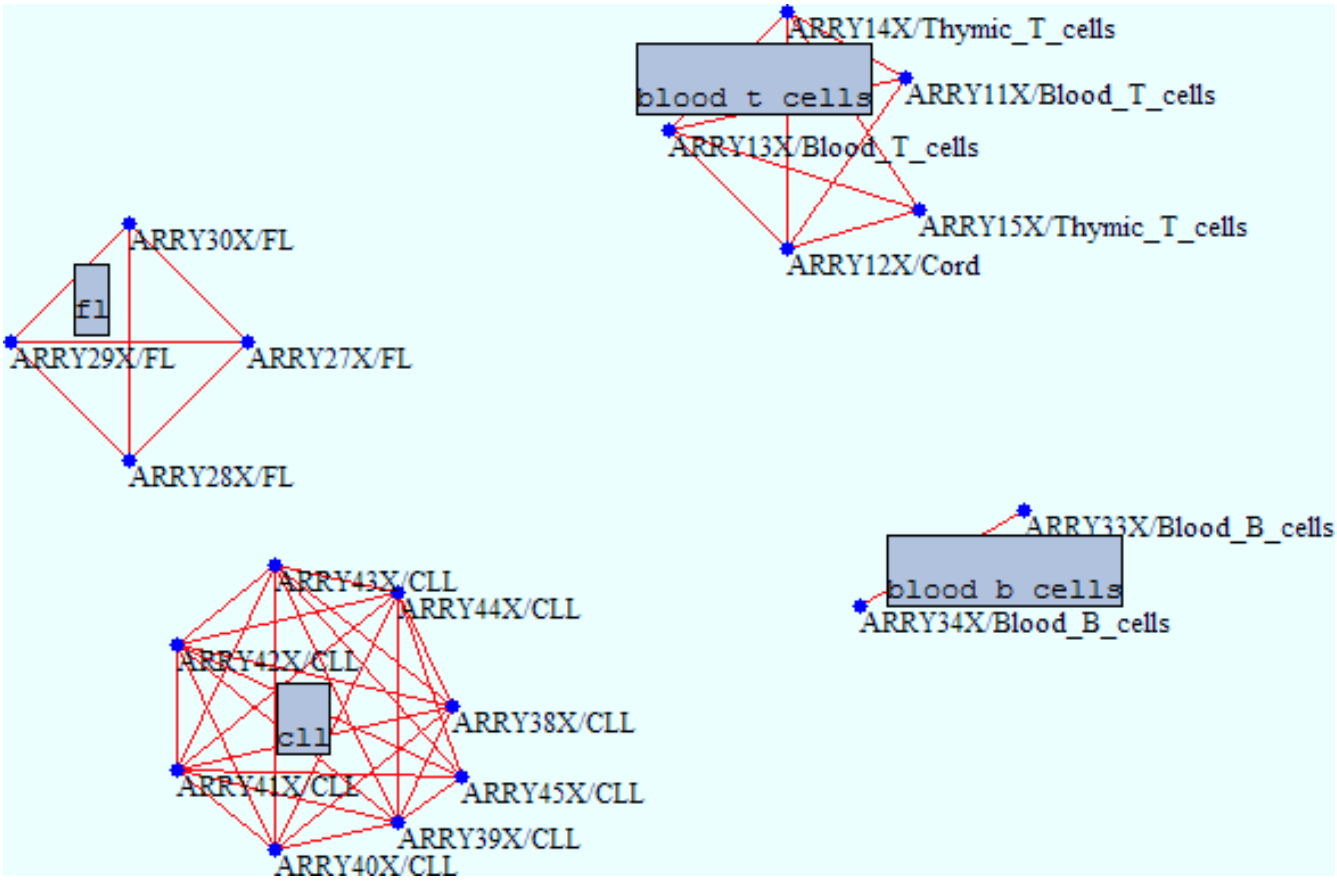

Figure 8
Clustering of the lymphoma samples with T_{cor} = 0.60 and T_{curv} = 0.30. Each cluster is associated with a cell type indicated in the grey boxes

Further Developments

Future development should include finer statistical analysis tools to validate the automatic annotations. In particular a bootstrap validation of a discriminant score [9] would be more accurate that the correlation score explained in the methods, which detects consistency with the annotation rather than actual discrimination. Also, more sophisticated methods for determining optimal annotations exist in the literature and could be applied to our clusters (see *e.g.* [11]).

A method for determining a natural correlation threshold T_{cor} would be most welcome (such methods have been discussed in [26,27]). It would leave only one free parameter, the curvature threshold T_{curv} Again a bootstrap calculation could provide an estimate of a significant deviation from average random correlation. It was also suggested to use a hierarchical construction of the graph: first use the strongest links (largest correlations) to build small clusters, then link clusters with weaker links and continue until all nodes belong to the same cluster. Varying T_{curv} would subsequently split this unique cluster into significant parts. Memory and speed limitations may hamper these developments.

Authors' Contributions

JR conceived of the study and carried out all programming as a postdoc in the group of PH. PH supervised the study and provided valuable input for the analyses and biological interpretations. Both authors read and approved the final manuscript.

Acknowledgments

We thank D. Gautheret for critically reading a first version of the manuscript. J.R. is grateful to J.-P. Eckmann and E. Moses for useful comments on earlier versions of the program and for detailed explanations of their results. Part of this research was supported by the Temblor project EU grant QLRT-2001-00015.

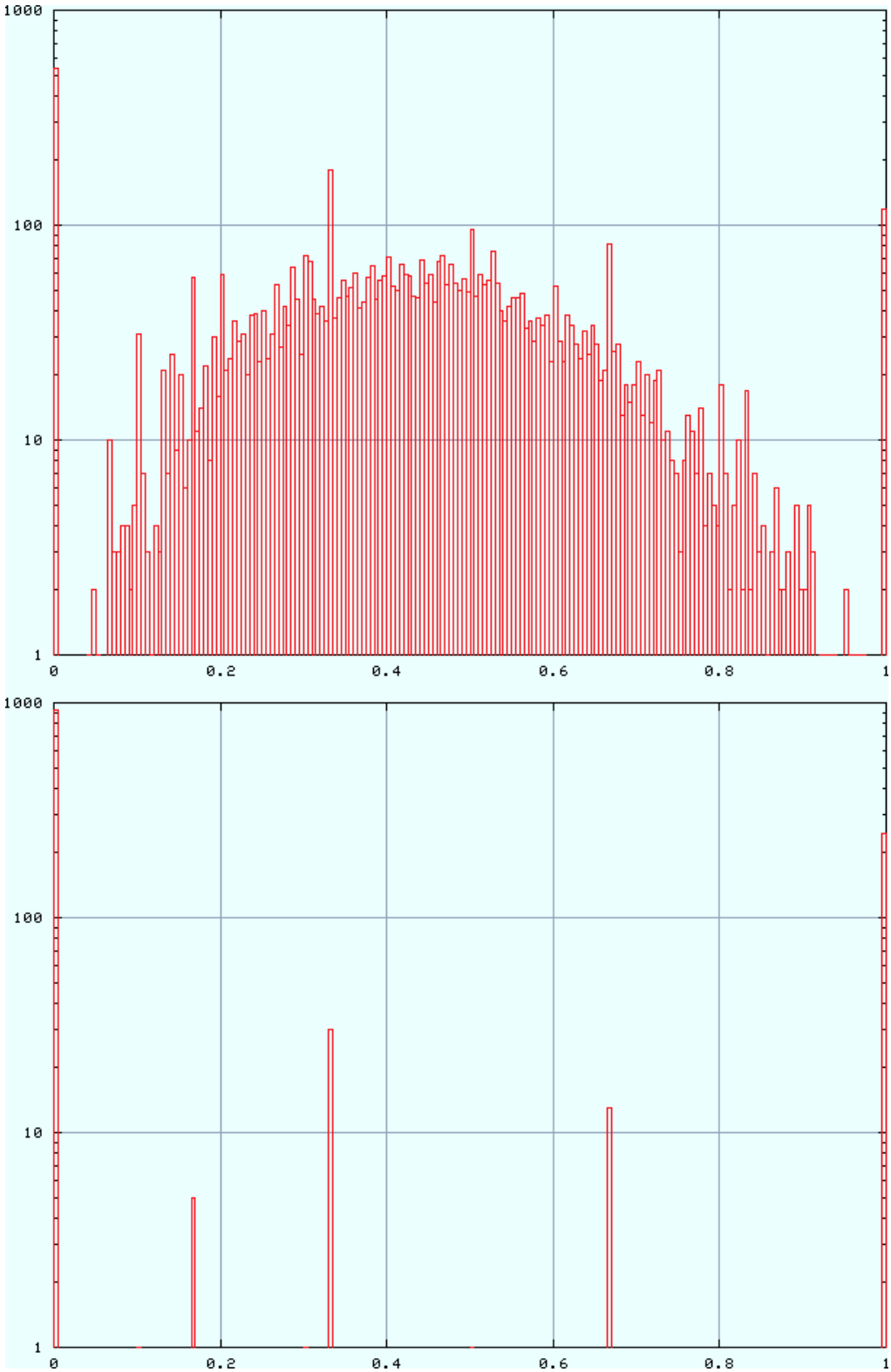

Figure 9
Distribution of curvatures in the yeast (top) and randomised yeast (bottom) data with T_{cor} = 0.6. Top: the mean curvature is 0.387 and 85% of nodes have positive curvature, bottom: cumulative distribution of curvatures after 1000 permutations in each of the 6221 genes. The proportion of nodes with non-zero curvature is 5×10^{-5}

References

1. Eisen MB, Spellman PT, Brown PO and Botstein D **Cluster analysis and display of genome-wide expression patterns** *Proc NatL Acad Sci USA* 1998, **95:**14863-14828
2. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES and Golub TR **Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation** *Proc Natl Acad Sci USA* 1999, **96:**2907-2912
3. Tavazoie S, Hughes JD, Campbell MJ, Cho RJ and Church GM **Systematic determination of genetic network architecture** *Nature Genetics* 1999, **22:**281-285
4. Gasch AP and Eisen MB **Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering** *Genome Biology* 2002, **3:**Research0059
5. Herrero J, Valencia A and Dopazo J **A hierarchical unsupervised growing neural network for clustering gene expression patterns** *Bioinformatics* 2001, **17:**126-136
6. Shamir R and Sharan R **Algorithmic approaches to clustering gene expression data** *In Current Topics In Computational Molecular Biology (Edited by: Jiang T, Xu Y, Smith T)* 2002, 269-300
7. Tamames J, Clark D, Herrero J, Dopazo J, Blaschke C, Fernandez JM, Oliveros JC and Valencia A **Bioinformatics methods for the analysis of expression arrays: data clustering and information extraction** *J Biotechnol* 2002, **98:**269-283
8. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M and Haussler D **Support vector machine classification and validation of cancer tissue samples using microarray data** *Bioinformatics* 2000, **16:**906-914
9. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR and Caligiuri MA **Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring** *Science* 1999, **286:**531-537
10. Wu LF, Hughes TR, Davierwala AP, Robinson MD, Stoughton R and Altschuler SJ **Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters** *Nature Genetics* 2002, **31:**255-265
11. Pe'er D, Regev A and Tanay A **Minreg: inferring an active regulator set** *Bioinformatics* 2002, **18:**s258-s267
12. Zhou X, Kao MC and Wong WH **Transitive functional annotation by shortest-path analysis of gene expression data** *Proc Natl Acad Sci USA* 2002, **99:**12783-12788
13. Eckmann JP and Moses E **Curvature of co-links uncovers hidden thematic layers in the world wide web** *Proc Natl Acad Sci USA* 2002, **99:**5825-5829
14. Shen-Orr SS, Milo R, Mangan S and Alon U **Network motifs in the transcriptional regulation network of *Escherichia coli*** *Nature Genetics* 2002, **31:**64-68
15. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D and Alon U **Network motifs: Simple building blocks of complex networks** *Science* 2002, **298:**824-827
16. Jenssen TK, Laegreid A, Komorowski J and Hovig E **A literature network of human genes for high-throughput analysis of gene expression** *Nat Gene* 2001, **38:**A21-28
17. Butte AJ, Tamayo P, Slonim D, Golub TR and Kohane IS **Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks** *Proc Natl Acad Sci USA* 2000, **97:**12182-12186
18. Yanai I and DeLisi C **The society of genes: networks of functional links between genes from comparative genomics** *Genome Biology* 2002, **3:**Research0064
19. Watts DJ and Strogatz SH **Collective dynamics of 'small-world' networks** *Nature* 1998, **393:**440-442
20. Collet P and Eckmann JP **The number of large graphs with a positive density of triangles** *J Stat Phys* 2002, **109:**923-943
21. Alizadeh AA, Eisen MB, Davis RE, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X and Powell JI **Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling** *Nature* 2000, **403:**503-522
22. Van Lint JH and Wilson RM **A course in combinatorics** *Cambridge, Cambridge University Press* 2001,
23. Dwight SS, Harris MA, Dolinski K, Ball CA, Binkley G, Christie KR, Fisk DG, Issel-Tarver L, Schroeder M and Sherlock G **Saccharomyces genome database (SGD) provides secondary gene annotation using the gene ontology (GO)** *Nucleic Acids Res* 2002, **30:**69-72
24. Jeong H, Tombor B, Albert R, Oltval ZN and Barabasi AL **The large-scale organization of metabolic networks** *Nature* 2000, **407:**651-654
25. Featherstone DE and Broadie K **Wrestling with pleiotropy: Genomic and topological analysis of the yeast gene expression network** *Bio Essays* 2002, **24:**267-274
26. Domany E **Cluster analysis of gene expression data** *J Stat Phys* 2003, **110:**1117-1139
27. Cheng Y and Church GM **Biclustering of expression data** *Proc Int Conf Intell Syst Mol Biol* 2000, **8:**93-103